11-14-00

A

JC962 U.S. PTO 11/13/00

PTO/SB/05 (11-00)
Approved for use through 10/31/2002. OMB 0651-0032
U S. Patent and Trademark Office; U S. DEPARTMENT OF COMMERCE

Please type a plus sign (+) inside this box ☐

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	1939-002
First Inventor	Bernardin
Title	Methods, Apparatus and ...
Express Mail Label No.	47167255US

JC541 U.S. PTO 09/711634 11/13/00

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☒ Fee Transmittal Form (e.g., PTO/SB/17) (Submit an original and a duplicate for fee processing)
2. ☒ Applicant claims small entity status. See 37 CFR 1.27.
3. ☒ Specification [Total Pages 41] (preferred arrangement set forth below)
   - Descriptive title of the invention
   - Cross Reference to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to sequence listing, a table, or a computer program listing appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings (if filed)
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 2]
5. Oath or Declaration [Total Pages 2]
   a. ☐ Newly executed (original or copy)
   b. ☐ Copy from a prior application (37 CFR 1.63 (d)) (for continuation/divisional with Box 18 completed)
      i. ☐ DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☐ Application Data Sheet. See 37 CFR 1.76
7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)
8. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
   a. ☐ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. ☐ CD-ROM or CD-R (2 copies); or
      ii. ☐ paper
   c. ☐ Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. ☐ Assignment Papers (cover sheet & document(s))
10. ☐ 37 CFR 3.73(b) Statement (when there is an assignee) ☐ Power of Attorney
11. ☐ English Translation Document (if applicable)
12. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
13. ☐ Preliminary Amendment
14. ☐ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
15. ☐ Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. ☐ Request and Certification under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. ☐ Other: ....

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:

☐ Continuation ☐ Divisional ☒ Continuation-in-part (CIP) of prior application No 09/583,244

Prior application information: Examiner ____________ Group Art Unit ____________

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

☐ Customer Number or Bar Code Label (Insert Customer No. or Attach bar code label here) or ☒ Correspondence address below

Name	David Garrod				
Address	Hopgood, Calimafde, Judlowe & Mondolino, LLP 60 E. 42nd St.				
City	New York	State	NY	Zip Code	10165
Country		Telephone	212-551-5000	Fax	212-949-2795

Name (Print/Type)	David Garrod	Registration No. (Attorney/Agent)	35,149
Signature	[signature]	Date	11/13/00

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231

Jc962 U.S. PTO
11/13/00

PTO/SB/17 (11-00)
Approved for use through 10/31/2002. OMB 0651-0032
U S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2001

Patent fees are subject to annual revision

TOTAL AMOUNT OF PAYMENT ($)

Complete if Known	
Application Number	
Filing Date	11/13/00
First Named Inventor	Bernardin
Examiner Name	
Group Art Unit	
Attorney Docket No.	1939-002

METHOD OF PAYMENT

1. [X] The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to

Deposit Account Number: 08-2776

Deposit Account Name: Hopgood, Calimafde

[X] Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

[X] Applicant claims small entity status. See 37 CFR 1.27

2. [ ] Payment Enclosed:
[ ] Check [ ] Credit card [ ] Money Order [ ] Other

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	710	201	355	Utility filing fee	355
106	320	206	160	Design filing fee	
107	490	207	245	Plant filing fee	
108	710	208	355	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($) 355

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	86	-20** =	66	X	9	=	594
Independent Claims	9	- 3** =	6	X	40	=	240
Multiple Dependent						=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	80	202	40	Independent claims in excess of 3
104	270	204	135	Multiple dependent claim, if not paid
109	80	209	40	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 834

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for *ex parte* reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	390	216	195	Extension for reply within second month	
117	890	217	445	Extension for reply within third month	
118	1,390	218	695	Extension for reply within fourth month	
128	1,890	228	945	Extension for reply within fifth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,240	241	620	Petition to revive - unintentional	
142	1,240	242	620	Utility issue fee (or reissue)	
143	440	243	220	Design issue fee	
144	600	244	300	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	130	123	130	Petitions related to provisional applications	
126	180	126	180	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	710	246	355	Filing a submission after final rejection (37 CFR § 1.129(a))	
149	710	249	355	For each additional invention to be examined (37 CFR § 1.129(b))	
179	710	279	355	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify) ____________________

*Reduced by Basic Filing Fee Paid **SUBTOTAL (3)** ($) 0

SUBMITTED BY				Complete *(if applicable)*	
Name *(Print/Type)*	David Garrod	Registration No. *(Attorney/Agent)*	35,149	Telephone	212-551-5000
Signature	[signature]			Date	

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

METHODS, APPARATUS AND ARTICLES-OF-MANUFACTURE FOR PROVIDING ALWAYS-LIVE DISTRIBUTED COMPUTING

By: James Bernardin and Peter Lee

CROSS-REFERENCE TO RELATED APPLICATION

This application is a continuation-in-part of U.S. Patent Application S/N 09/583,244, filed 5/31/00, by the inventors herein ("the '244 application"), which prior application is incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to the fields of distributed computing methods, computer-assisted business methods, and systems and articles-of-manufacture for implementing such methods. More particularly, the invention relates to computer-based methods, apparatus and articles-of-manufacture for providing "always-live" (i.e., substantially continuously active and uninterrupted) distributed computing services in a network-based computing environment.

BACKGROUND OF THE INVENTION

Methods for providing distributed computing in network-based computing environments (such as the Internet) are known. One widely-publicized effort was the so-called SETI@home (Search for Extra-Terrestrial Intelligence) project, in which large numbers of Internet-connected computers were used to

process radio-telescope data, in an effort to identify patterns indicative of intelligent life. Other examples are described in U.S. Patent Nos. 5,964,832 ("USING NETWORKED REMOTE COMPUTERS TO EXECUTE COMPUTER PROCESSING TASKS AT A PREDETERMINED TIME"), 6,098,091 ("METHOD AND SYSTEM INCLUDING A CENTRAL COMPUTER THAT ASSIGNS TASKS TO IDLE WORKSTATIONS USING AVAILABILITY SCHEDULES AND COMPUTATIONAL CAPABILITIES") and 6,112,243 ("METHOD AND APPARATUS FOR ALLOCATING TASKS TO REMOTE NETWORKED PROCESSORS"), all owned by Intel Corporation. Still another example is disclosed in the earlier-filed '244 application by the inventors herein. (Note, however, that the '244 application is <u>not</u> prior art to the present invention.)

Generally speaking, the primary object of Internet-based distributed computing systems is to exploit the vast computational resources that sit idle for much of the 24-hour day on computer networks around the world. Although some success has been achieved, prior-art systems still have problems that limit their usefulness in real-world applications.

One particularly-troublesome aspect of the prior-art systems is their inability guarantee timely results. While it may be no problem for the SETI@home researchers to wait days or weeks for results from a particular data set, commercial customers simply cannot afford to have overnight processing jobs run unexpectedly into the next business day. Therefore, in order to realize the full commercial

potential of network-based distributed computing, it is necessary to ensure that the clients' work gets processed in a substantially continuous and uninterrupted manner, so that a service provider can assure his/her client that assigned work will be completed in within a commercially-reasonable time period (e.g., an hour, four hours, eight hours, etc.).

OBJECTS AND DESCRIPTION OF THE INVENTION

In light of the above, a first general object of the invention relates to computer-based methods, apparatus and articles-of-manufacture that facilitate an always-live distributed computing system.

A second general object of the invention relates to computer-based methods, apparatus and articles-of-manufacture that provide substantially continuous monitoring of worker processor activity and/or task progress in a distributed computing environment.

A third general object of the invention relates to computer-based methods, apparatus and articles-of-manufacture that provide prompt alerts of worker processor status changes that can affect the always-live operation of a network-based distributed computing system.

A fourth general object of the invention relates to computer-based methods, apparatus and articles-of-manufacture for providing reliable and/or predictable resource deployment and processing activity in a wide-area network

based distributed computing system.

These, as well as other objects and advantages of the present invention, will become apparent in light of the following description, which details, by way of example, various aspects and features of the present invention.

Accordingly, generally speaking, and without intending to be limiting, one aspect of the invention relates to a method for operating a distributed computing system, the system including a multiplicity of network-connected worker processors and at least one supervisory processor, the supervisory processor configured to assign tasks to, and monitor the status of, the worker processors, the method comprising: assigning tasks to a plurality of the worker processors by sending task-assignment messages, via the network, from the at least one supervisory processor to the plurality of worker processors; and monitoring, on a substantially continuous basis, the status of at least each of the plurality of assigned worker processors until each processor completes its assigned task. Monitoring, on a substantially continuous basis, the status of at least each of the plurality of assigned worker processors may involve receiving status messages from at least each of the plurality of assigned worker processors until each processor completes its assigned task. Monitoring, on a substantially continuous basis, the status of at least each of the plurality of worker processors may also involve detecting abnormalities in the operation of the plurality of assigned worker processors, and/or their associated network connections, by detecting an absence of expected status message(s)

received by the at least one supervisory processor. Detection of an absence of expected status message(s) received by the at least one supervisory processor may be repeated at least once every ten minutes, once every five minutes, once every two minutes, once each minute, once every thirty seconds, once every ten seconds, once every second, once every tenth of a second, once every hundredth of a second, once each millisecond, or at whatever interval is needed to assure the continuity-of-service demanded by the client. Monitoring, on a substantially continuous basis, the status of at least each of the plurality of assigned worker processors may also involve detecting the presence of non-assigned-task-related activity on the worker processors. Detecting the presence of non-assigned-task-related activity on the worker processors may involve running an activity monitor program on each of the assigned worker processors. The activity monitor programs running on each of the assigned worker processors may behave substantially like screen saver programs. The activity monitory programs running on each of the assigned worker processors may send, in response to detection of keyboard activity (or mouse activity, pointer activity, touchscreen activity, voice activity, local execution of substantial non-assigned-task-related processes, or any combination thereof), a message to at least one of the at least one supervisory processor(s). Detecting the presence of non-assigned-task-related activity on the worker processors may also involve determining, in response to an activity monitor message received by at least one of the at least one supervisory of the processor(s),

that at least one of the assigned worker processors is undertaking non-assigned-task-related activity. The activity monitor message may be generated by an activity monitor program running on one of the assigned worker processors.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to a method for operating an always-live distributed computing system, comprising: providing a pool of worker processors, each having installed worker processor software, and each connected to an always-on, peer-to-peer computer network; providing at least one supervisory processor, also connected to the always-on, peer-to-peer computer network; using the at least one supervisory processor to monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks; and using the at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks. Providing a pool of worker processors may further involve ensuring that each of the worker processors is linked to the always-on, peer-to-peer computer network through a high-bandwidth connection having, for example, a data rate of least 100 kilobits/sec, 250 kilobits/sec, 1 megabit/sec, 10 megabits/sec, 100 megabits/sec, 1 gigabit/sec, or whatever particular bandwidth may be demanded by the client's needs (e.g., required throughput and data intensiveness of the application). Using the at least one supervisory processor to monitor the status of worker processors expected to be engaged in the processing of assigned tasks may involve sending a

status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks. The process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is preferably repeated at least once every minute, second, tenth of a second, hundredth of a second, millisecond or other interval, as needed to meet client requirements. Using the at least one supervisory processor to monitor the status of worker processors expected to be engaged in the processing of assigned tasks may also involve periodically checking to ensure that a heartbeat message has been received, within a preselected frequency interval, from each worker processor that is expected to be engaged in the processing of assigned tasks. The preselected frequency interval may be set at or less than one minute, ten seconds, one second, one tenth of a second, one hundredth of a second, one millisecond, or other appropriate value, as needed. Using the at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks may also involve: detecting aberrant behavior among the worker processors expected to be engaged in the processing of assigned tasks; and reassigning tasks expected to be completed by the aberrant-behaving worker processor(s) to other available processor(s) in the worker processor pool.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to a method for operating a network-connected processor as a processing element in a distributed processing system, the method comprising: installing software that enables the network-connected processor to receive tasks from, and provide results to, one or more independent, network-connected resource(s); and using the software installed on the network-connected processor to provide substantially continuous status information to an independent, network-connected resource. Using the software installed on the network-connected processor to provide substantially continuous status information to an independent, network-connected resource may involve sending a heartbeat message to the independent, network-connected resource at least once every second, tenth of a second, hundredth of a second, millisecond, etc. Using the software installed on the network-connected processor to provide substantially continuous status information to an independent, network-connected resource may also involve responding to status-request messages, received from the independent, network-connected resource, within a predetermined response time, such as one second, one tenth of a second, one hundredth of a second, one millisecond, etc. Using the software installed on the network-connected processor to provide substantially continuous status information to an independent, network-connected resource may also involve sending, in response to a change in status of the network-connected processor, a status-update message to the independent,

network-connected resource within a preselected update interval, such as one second, one tenth of a second, one hundredth of a second, one millisecond, etc. The change in status that initiates the sending of a status-update message may include any local activity indicator (such as keyboard activity, other processes in the process queue, etc.) that indicates additional demand for the processing resources of the network-connected processor.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to a distributed computing system comprising: a multiplicity of worker processors; at least one supervisory processor, configured to assign tasks to, and monitor the status of, the worker processors; an always-on, peer-to-peer computer network linking the worker processors and the supervisory processor(s); and at least one of the at least one supervisory processor(s) including a monitoring module, which monitors the status of worker processors expected to be executing assigned tasks, so as to ensure that the distributed computing system maintains always-live operation. The monitoring module may receive status messages from at least each of the worker processors expected to be executing assigned tasks. The monitoring module may be used to detect abnormalities in the operation of the worker processors expected to be executing assigned tasks, and/or their associated network connections, by, for example, detecting an absence of expected status messages received from the worker processors. The monitoring module may repeatedly check for an absence of expected status messages at a

frequency of at least once each minute, at least once every ten seconds, at least once each second, at least once every tenth of a second, etc. The monitoring module may also be used to detect the presence of non-assigned-task-related activity on the worker processors expected to be executing assigned tasks. Activity monitor programs may be run on each of the worker processors expected to be executing assigned tasks. The activity monitor programs comprise screensaver programs. The activity monitor programs may be configured to detect one or more of the following types of non-assigned-task-related activity: keyboard activity; mouse activity; pointer activity; touchscreen activity; voice activity; and execution of substantial non-assigned-task-related processes.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to an always-live distributed computing system, comprising: a pool of worker processors, each having installed worker processor software, and each connected to an always-on, peer-to-peer computer network; and at least one supervisory processor, also connected to the always-on, peer-to-peer computer network, and configured to assign tasks to the worker processors, monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks and reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks. The computer network may have a bandwidth of at least 250 kilobits/second, at least 1 megabit/second, etc. The at least one supervisory processor may monitor the status

of worker processors expected to be engaged in the processing of assigned tasks by sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks. Such status-request message(s) may be sent at a frequency of at least once every 10 seconds, at least once each second, at least twenty times each second, etc. The at least one supervisory processor may monitor the status of worker processors expected to be engaged in the processing of assigned tasks by periodically checking to ensure that a heartbeat message has been received, within a preselected frequency interval, from each worker processor that is expected to be engaged in the processing of assigned tasks. The preselected frequency interval may be, for example, one second, one tenth of a second, one hundredth of a second, one millisecond, etc.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to a processing element for use in a distributed processing system, the processing element comprising: at least one processor; memory; at least one high-bandwidth interface to a computer network; and worker processor software, configured to receive tasks via the high-bandwidth interface and to provide substantially continuous status information via the high-bandwidth interface. The substantially continuous status information may be provided by sending periodic heartbeat messages. The substantially continuous status information may also be provided by sending prompt responses to received

status-request messages. The substantially continuous status information may also be provided by promptly sending a status-update message in response to changes in status.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to article(s)-of-manufacture for use in connection with a network-based distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause: assignment of tasks to a plurality of worker processors via the network; and monitoring, on a substantially continuous basis, of the status of at least each of the plurality of assigned worker processors until each such processor completes its assigned task.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to article(s)-of-manufacture for use in connection with an always-live distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause: a pool of worker processors to install worker processor software provided via an always-on, peer-to-peer computer network; provide communication paths between the worker processors and at least one supervisory processor via the always-on, peer-to-peer computer network; cause the at least one supervisory processor to monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks; and

cause the at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to article(s)-of-manufacture for use in connection with a processing element constituting a part of a distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause: worker processor software to be installed that permits the processing element to receive tasks from, and provide results to, one or more independent, network-connected resource(s); and the installed worker processor software to be executed and provide substantially continuous status information to one or more of the independent, network-connected resource(s).

Still further aspects of the invention relate to alternative combinations, sub-combinations, supplemental combinations and/or permutations of the various above-described elements and features, as well as those elements and features described in the incorporated '244 application, consistent with or in furtherance of the objects and spirit of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Various aspects, features and advantages of the instant invention are depicted in the accompanying set figures, which is intended to be illustrative, rather

than limiting, and in which:

FIG. 1 depicts an exemplary network-based distributed processing system in which the present invention may be employed; and,

FIG. 2 contains a flowchart illustrating the operation of an exemplary always-live distributed processing system in accordance with the invention.

DESCRIPTION OF AN EXEMPLARY EMBODIMENT

Referring initially to FIG. 1, which depicts an exemplary context in which the method(s), apparatus and/or article(s)-of-manufacture of the invention may be applied, a computer network **1** is shown connecting a plurality of processing resources. (Although, for clarity, only six processing resources are shown in FIG. 1, the invention is preferably deployed in networks connecting hundreds, thousands, tens of thousands or greater numbers of processing resources.) Computer network **1** may utilize any type of transmission medium (e.g., wire, coax, fiber optics, RF, satellite, etc.) and any network protocol. However, in order to realize the principal benefit(s) of the present invention, computer network **1** should provide a relatively high bandwidth (e.g., at least 100 kilobits/second) and preferably, though not necessarily, should provide an "always on" connection to the processing resources involved in distributed processing activities.

Still referring to FIG. 1, one or more supervisory processor(s) **13** may communicate with a plurality of worker processors **10** via computer network **1**. Supervisory processor(s) **13** perform such tasks as:

- accepting job(s) from clients;
- assigning/reassigning tasks to (or among) worker processors;
- managing pools of available worker processors;
- monitoring the status of worker processors;
- monitoring the status of network connections;
- monitoring the status of job and task completions; and/or,
- resource utilization tracking, timekeeping and billing.

Still referring to FIG. 1, the depicted plurality **13** of worker processors **11** and **12** may operate collaboratively as a group, independently (e.g., each handing different job(s), task(s) and/or worker processor pool(s)) and/or redundantly (thus providing enhanced reliability). However, to realize a complete distributed processing system in accordance with the invention, only a single supervisory processor (e.g., **11** or **12**) is needed.

Still referring to FIG. 1, plurality **10** of worker processors illustratively comprises worker processors **2**, **4**, **6** and **8**, each connected to computer network **1** through network connections **3**, **5**, **7** and **9**, respectively. These worker processors communicate with supervisory processor(s) **13** via network **1**, and preferably include worker processor software that enables substantially continuous monitoring of

worker processor status and/or task execution progress by supervisory processor(s) **13**.

Referring now to FIG. 2, which depicts an exemplary "always-live" task monitoring/management process, a received job request **20** is initially assigned **21** to a plurality of available worker processors. Then, until the client's job is completed, processor(s) working on assigned task(s) are continuously monitored to ensure that the job is completed in a substantially uninterrupted (or "always live") manner. In particular, a monitoring module repeatedly asks whether all assigned tasks have been completed **22**. If so, then the job is complete, and results can be reported **23**. If not, then the monitoring module inquires about the status **24** of processor(s) expected to be working on not-yet-completed tasks. If potential bottlenecks are discovered, affected task(s) are immediately reassigned **25** to ensure that the system remains "live" and the client's work gets completed in a timely manner. This process is repeated with a frequency sufficient to ensure that worker processor problems will not cause undue delay is completing the overall job.

While the foregoing has described the invention by recitation of its various aspects/features and an illustrative embodiment thereof, those skilled in the art will recognize that alternative elements and techniques, and/or combinations and sub-combinations of the described elements and techniques, can be substituted for, or added to, those described herein. The present invention, therefore, should not be limited to, or defined by, the specific apparatus, methods, and articles-of-

manufacture described herein, but rather by the appended claims, which are intended to be construed in accordance with well-settled principles of claim construction, including, but not limited to, the following:

Limitations should not be read from the specification or drawings into the claims (e.g., if the claim calls for a "chair," and the specification and drawings show a rocking chair, the claim term "chair" should not be limited to a rocking chair, but rather should be construed to cover any type of "chair").

The words "comprising," "including," and "having" are always open-ended, irrespective of whether they appear as the primary transitional phrase of a claim, or as a transitional phrase within an element or sub-element of the claim (e.g., the claim "a widget comprising: A; B; and C" would be infringed by a device containing 2A's, B, and 3C's; also, the claim "a gizmo comprising: A; B, including X, Y, and Z; and C, having P and Q" would be infringed by a device containing 3A's, 2X's, 3Y's, Z, 6P's, and Q).

The indefinite articles "a" or "an" mean "one or more"; where, instead, a purely singular meaning is intended, a phrase such as "one," "only one," or "a single," will appear.

Where the phrase "means for" precedes a data processing or

manipulation "function," it is intended that the resulting means-plus-function element be construed to cover any, and all, computer implementation(s) of the recited "function" using any standard programming techniques known by, or available to, persons skilled in the computer programming arts.

A claim that contains more than one computer-implemented means-plus-function element should not be construed to require that each means-plus-function element must be a structurally distinct entity (such as a particular piece of hardware or block of code); rather, such claim should be construed merely to require that the overall combination of hardware/software which implements the invention must, as a whole, implement at least the function(s) called for by the claims.

WHAT WE CLAIM IS:

1. A method for operating a distributed computing system, said system including a multiplicity of network-connected worker processors and at least one supervisory processor, said supervisory processor configured to assign tasks to, and monitor the status of, said worker processors, said method comprising:

assigning tasks to a plurality of said worker processors by sending task-assignment messages, via said network, from said at least one supervisory processor to said plurality of worker processors; and,

monitoring, on a substantially continuous basis, the status of at least each of said plurality of assigned worker processors until each said processor completes its assigned task.

2. A method for operating a distributed computing system, as defined in claim 1, wherein monitoring, on a substantially continuous basis, the status of at least each of said plurality of assigned worker processors comprises receiving status messages from at least each of said plurality of assigned worker processors until each said processor completes its assigned task.

3. A method for operating a distributed computing system, as defined in claim 2, wherein monitoring, on a substantially continuous basis, the status of at least each of said plurality of worker processors further comprises detecting abnormalities in the operation of said plurality of assigned worker processors, and/or

their associated network connections, by detecting an absence of expected status message(s) received by said at least one supervisory processor.

4. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once every ten minutes.

5. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once every five minutes.

6. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once every two minutes.

7. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once each minute.

8. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once every

thirty seconds.

9. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once every ten seconds.

10. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once every second.

11. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once every tenth of a second.

12. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once every hundredth of a second.

13. A method for operating a distributed computing system, as defined in claim 3, wherein said act of detecting an absence of expected status message(s) received by said at least one supervisory processor is repeated at least once each millisecond.

14. A method for operating a distributed computing system, as defined in claim 1, wherein monitoring, on a substantially continuous basis, the status of at least each of said plurality of assigned worker processors comprises:

detecting the presence of non-assigned-task-related activity on said worker processors.

15. A method for operating a distributed computing system, as defined in claim 14, wherein detecting the presence of non-assigned-task-related activity on said worker processors includes:

running an activity monitor program on each of said assigned worker processors.

16. A method for operating a distributed computing system, as defined in claim 15, wherein:

the activity monitor programs running on each of said assigned worker processors behave substantially like screen saver programs.

17. A method for operating a distributed computing system, as defined in claim 15, wherein:

the activity monitory programs running on each of said assigned worker processors send, in response to detection of keyboard activity, a message to at least one of said at least one supervisory processor(s).

18. A method for operating a distributed computing system, as defined in claim 15, wherein:

the activity monitory programs running on each of said assigned worker processors send, in response to detection of mouse activity, a message to at least one of said at least one supervisory processor(s).

19. A method for operating a distributed computing system, as defined in claim 15, wherein:

the activity monitory programs running on each of said assigned worker processors send, in response to detection of pointer activity, a message to at least one of said at least one supervisory processor(s).

20. A method for operating a distributed computing system, as defined in claim 15, wherein:

the activity monitory programs running on each of said assigned worker processors send, in response to detection of touchscreen activity, a message to at least one of said at least one supervisory processor(s).

21. A method for operating a distributed computing system, as defined in claim 15, wherein:

the activity monitory programs running on each of said assigned worker

processors send, in response to detection of voice activity, a message to at least one of said at least one supervisory processor(s).

22. A method for operating a distributed computing system, as defined in claim 15, wherein:

the activity monitory programs running on each of said assigned worker processors send, in response to detection of execution of substantial non-assigned-task-related processes, a message to at least one of said at least one supervisory processor(s).

23. A method for operating a distributed computing system, as defined in claim 14, wherein detecting the presence of non-assigned-task-related activity on said worker processors includes:

determining, in response to an activity monitor message received by at least one of said at least one supervisory of said processor(s), that at least one of said assigned worker processors is undertaking non-assigned-task-related activity.

24. A method for operating a distributed computing system, as defined in claim 23, wherein the activity monitor message is generated by an activity monitor program running on one of said assigned worker processors.

25. A method for operating an always-live distributed computing system, comprising:

providing a pool of worker processors, each having installed worker processor software, and each connected to an always-on, peer-to-peer computer network;

providing at least one supervisory processor, also connected to said always-on, peer-to-peer computer network;

using said at least one supervisory processor to monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks; and,

using said at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks.

26. A method for operating an always-live distributed computing system, as defined in claim 25, wherein providing a pool of worker processors further includes ensuring that each of said worker processors is linked to said always-on, peer-to-peer computer network through a high-bandwidth connection.

27. A method for operating an always-live distributed computing system, as defined in claim 25, wherein providing a pool of worker processors further includes ensuring that each of said worker processors is linked to said always-on, peer-to-peer computer network at a data rate of at least 100 kilobits/sec.

28. A method for operating an always-live distributed computing system, as defined in claim 25, wherein providing a pool of worker processors further includes ensuring that each of said worker processors is linked to said always-on, peer-to-peer computer network at a data rate of at least 250 kilobits/sec.

29. A method for operating an always-live distributed computing system, as defined in claim 25, wherein providing a pool of worker processors further includes ensuring that each of said worker processors is linked to said always-on, peer-to-peer computer network at a data rate of at least 1 megabit/sec.

30. A method for operating an always-live distributed computing system, as defined in claim 25, wherein providing a pool of worker processors further includes ensuring that each of said worker processors is linked to said always-on, peer-to-peer computer network at a data rate of at least 10 megabits/sec.

31. A method for operating an always-live distributed computing system, as defined in claim 25, wherein providing a pool of worker processors further includes ensuring that each of said worker processors is linked to said always-on, peer-to-peer computer network at a data rate of at least 100 megabits/sec.

32. A method for operating an always-live distributed computing system, as defined in claim 25, wherein providing a pool of worker processors further includes ensuring that each of said worker processors is linked to said always-on, peer-to-peer computer network at a data rate of at least 1 gigabit/sec.

33. A method for operating an always-live distributed computing system, as defined in claim 25, wherein using said at least one supervisory processor to monitor the status of worker processors expected to be engaged in the processing of assigned tasks includes:

sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks.

34. A method for operating an always-live distributed computing system, as defined in claim 33, wherein said process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is repeated at least once every second.

35. A method for operating an always-live distributed computing system, as defined in claim 33, wherein said process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is repeated at least once every tenth of a second.

36. A method for operating an always-live distributed computing system, as defined in claim 33, wherein said process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is repeated at least

once every hundredth of a second.

37. A method for operating an always-live distributed computing system, as defined in claim 33, wherein said process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is repeated at least once every millisecond.

38. A method for operating an always-live distributed computing system, as defined in claim 25, wherein using said at least one supervisory processor to monitor the status of worker processors expected to be engaged in the processing of assigned tasks includes:

periodically checking to ensure that a heartbeat message has been received, within a preselected frequency interval, from each worker processor that is expected to be engaged in the processing of assigned tasks.

39. A method for operating an always-live distributed computing system, as defined in claim 38, wherein said preselected frequency interval is less than one second.

40. A method for operating an always-live distributed computing system, as defined in claim 38, wherein said preselected frequency interval is less than one tenth of a second.

41. A method for operating an always-live distributed computing system, as defined in claim 38, wherein said preselected frequency interval is less than one hundredth of a second.

42. A method for operating an always-live distributed computing system, as defined in claim 38, wherein said preselected frequency interval is less than one millisecond.

43. A method for operating an always-live distributed computing system, as defined in claim 25, wherein using said at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks comprises:

detecting aberrant behavior among the worker processors expected to be engaged in the processing of assigned tasks; and,

assigning tasks expected to be completed by said aberrant-behaving worker processor(s) to other available processor(s) in said worker processor pool.

44. A method for operating a network-connected processor as a processing element in a distributed processing system, the method comprising:

installing software that enables said network-connected processor to receive tasks from, and provide results to, one or more independent, network-connected resource(s); and,

using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource.

45. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

sending a heartbeat message to said independent, network-connected resource at least once every second.

46. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

sending a heartbeat message to said independent, network-connected resource at least once every tenth of a second.

47. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource

includes:

sending a heartbeat message to said independent, network-connected resource at least once every hundredth of a second.

48. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

sending a heartbeat message to said independent, network-connected resource at least once every millisecond.

49. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

responding to status-request messages, received from said independent, network-connected resource, within one second.

50. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource

includes:

responding to status-request messages, received from said independent, network-connected resource, within one tenth of a second.

51. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

responding to status-request messages, received from said independent, network-connected resource, within one hundredth of a second.

52. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

responding to status-request messages, received from said independent, network-connected resource, within one millisecond.

52. A method for operating a network-connected processor as a processing

element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

sending, in response to a change in status of said network-connected processor, a status-update message to said independent, network-connected resource within one second.

53. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

sending, in response to a change in status of said network-connected processor, a status-update message to said independent, network-connected resource within one tenth of a second.

54. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 44, wherein using the software installed on said network-connected processor to provide substantially continuous status information to an independent, network-connected resource includes:

sending, in response to a change in status of said network-connected processor, a status-update message to said independent, network-connected resource within one hundredth of a second.

55. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 52, wherein the change in status that initiates the sending of a status-update message is additional demand for the processing resources of the network-connected processor.

56. A method for operating a network-connected processor as a processing element in a distributed processing system, as defined in claim 52, wherein the change in status that initiates the sending of a status-update message is user input-related activity on the network-connected processor.

57. A distributed computing system comprising:

a multiplicity of worker processors;

at least one supervisory processor, configured to assign tasks to, and monitor the status of, said worker processors;

an always-on, peer-to-peer computer network linking said worker processors and said supervisory processor(s); and,

at least one of said at least one supervisory processor(s) including a monitoring module, which monitors the status of worker processors expected to be executing assigned tasks, so as to ensure that the distributed computing system maintains always-

live operation.

58. A distributed computing system, as defined in claim 57, wherein the monitoring module receives status messages from at least each of the worker processors expected to be executing assigned tasks.

59. A distributed computing system, as defined in claim 58, wherein the monitoring module detects abnormalities in the operation of said worker processors expected to be executing assigned tasks, and/or their associated network connections, by detecting an absence of expected status messages received from said worker processors.

60. A distributed computing system, as defined in claim 59, wherein the monitoring module checks for an absence of expected status messages at least once each minute.

61. A distributed computing system, as defined in claim 59, wherein the monitoring module checks for an absence of expected status messages at least once every ten seconds.

62. A distributed computing system, as defined in claim 59, wherein the monitoring module checks for an absence of expected status messages at least once each second.

63. A distributed computing system, as defined in claim 59, wherein the monitoring module checks for an absence of expected status messages at least once every tenth of a second.

64. A distributed computing system, as defined in claim 57, wherein the monitoring module detects the presence of non-assigned-task-related activity on the worker processors expected to be executing assigned tasks.

65. A distributed computing system, as defined in claim 64, further comprising:

activity monitor programs running on each of the worker processors expected to be executing assigned tasks.

66. A distributed computing system, as defined in claim 65, wherein the activity monitor programs comprise screensaver programs.

67. A distributed computing system, as defined in claim 64, wherein the activity monitor programs detect at least one of the following types of non-assigned-task-related activity:

keyboard activity;

mouse activity;

pointer activity;

touchscreen activity;

voice activity; and,

execution of substantial non-assigned-task-related processes.

68. A distributed computing system, as defined in claim 64, wherein the activity monitor programs detect at least three of the following types of non-assigned-task-related activity:

keyboard activity;

mouse activity;

pointer activity;

touchscreen activity;

voice activity; and,

execution of substantial non-assigned-task-related processes.

69. An always-live distributed computing system, comprising:

a pool of worker processors, each having installed worker processor software, and each connected to an always-on, peer-to-peer computer network; and,

at least one supervisory processor, also connected to said always-on, peer-to-peer computer network, and configured to assign tasks to said worker processors, monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks and reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks.

70. An always-live distributed computing system, as defined in claim 69, wherein said computer network has a bandwidth of at least 250 kilobits/second.

71. An always-live distributed computing system, as defined in claim 69, wherein said computer network has a bandwidth of at least 1 megabit/second.

72. An always-live distributed computing system, as defined in claim 69, wherein the at least one supervisory processor monitors the status of worker processors expected to be engaged in the processing of assigned tasks by sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks.

73. An always-live distributed computing system, as defined in claim 69, wherein the process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is repeated at least once every 10 seconds.

74. An always-live distributed computing system, as defined in claim 69, wherein the process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is repeated at least once each second.

75. An always-live distributed computing system, as defined in claim 69, wherein the process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is repeated at least twenty times each second.

76. An always-live distributed computing system, as defined in claim 69, wherein the at least one supervisory processor monitors the status of worker processors expected to be engaged in the processing of assigned tasks by

periodically checking to ensure that a heartbeat message has been received, within a preselected frequency interval, from each worker processor that is expected to be engaged in the processing of assigned tasks.

77. An always-live distributed computing system, as defined in claim 76, wherein the preselected frequency interval is less than one second.

78. An always-live distributed computing system, as defined in claim 76, wherein the preselected frequency interval is less than one tenth of a second.

79. An always-live distributed computing system, as defined in claim 76, wherein the preselected frequency interval is less than one hundredth of a second.

80. A processing element for use in a distributed processing system, the processing element comprising:

at least one processor;

memory;

at least one high-bandwidth interface to a computer network; and,

worker processor software, configured to receive tasks via said high-bandwidth interface and to provide substantially continuous status information via said high-bandwidth interface.

81. A processing element, as defined in claim 80, wherein substantially continuous status information is provided by sending periodic heartbeat messages.

82. A processing element, as defined in claim 80, wherein substantially continuous status information is provided by sending prompt responses to received

status-request messages.

83. A processing element, as defined in claim 80, wherein substantially continuous status information is provided by promptly sending a status-update message in response to a change in status.

84. Article(s)-of-manufacture for use in connection with a network-based distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause:

assignment of tasks to a plurality of worker processors via said network; and,

monitoring, on a substantially continuous basis, of the status of at least each of said plurality of assigned worker processors until each said processor completes its assigned task.

85. Article(s)-of-manufacture for use in connection with an always-live distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause:

a pool of worker processors to install worker processor software provided via an always-on, peer-to-peer computer network;

provide communication paths between said worker processors and at least one supervisory processor via said always-on, peer-to-peer computer network;

cause said at least one supervisory processor to monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks; and,

cause said at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks.

86. Article(s)-of-manufacture for use in connection with a processing element constituting a part of a distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause:

worker processor software to be installed that permits said processing element to receive tasks from, and provide results to, one or more independent, network-connected resource(s); and,

said installed worker processor software to be executed and provide substantially continuous status information to one or more of said independent, network-connected resource(s).

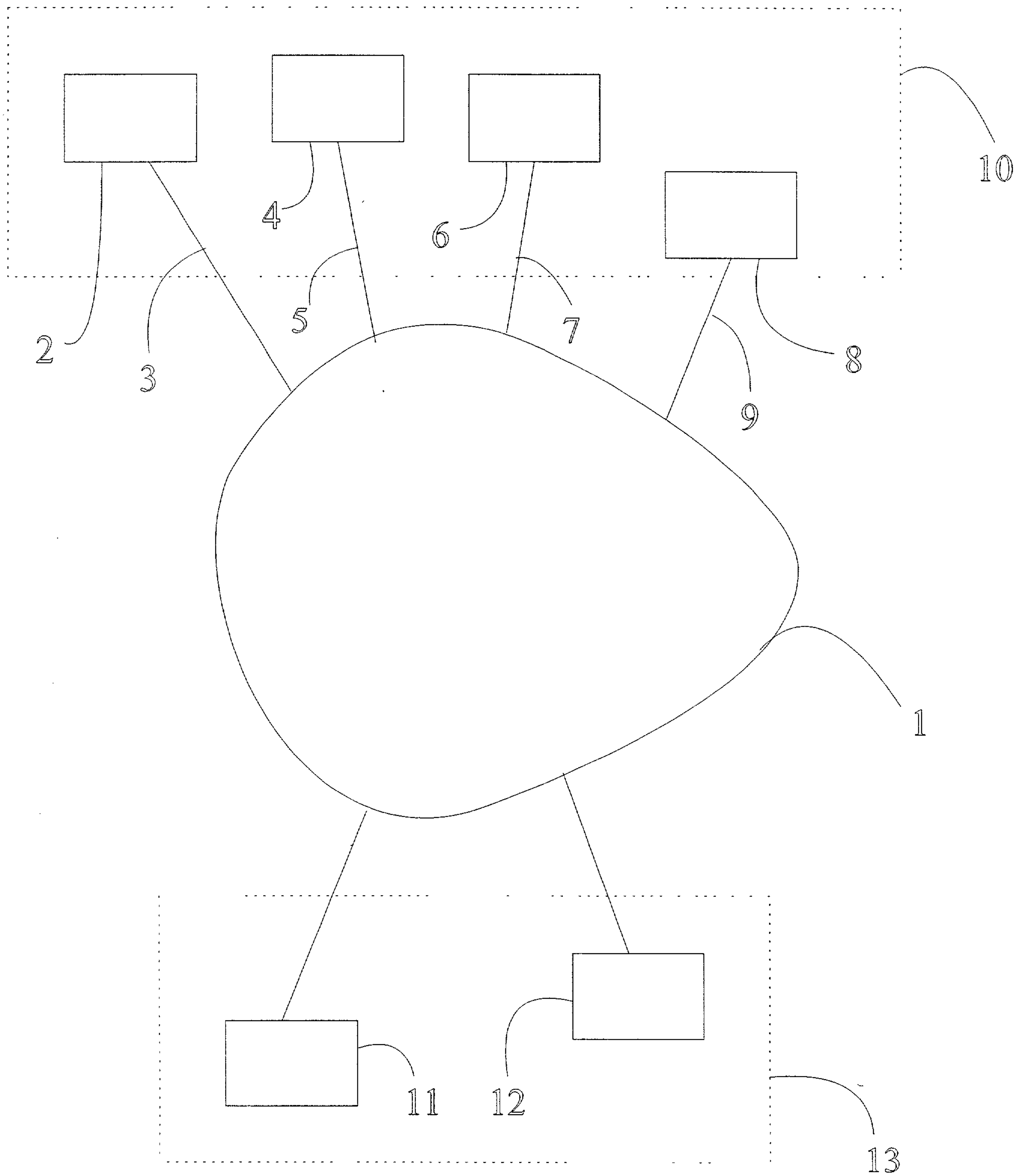

Fig. 1

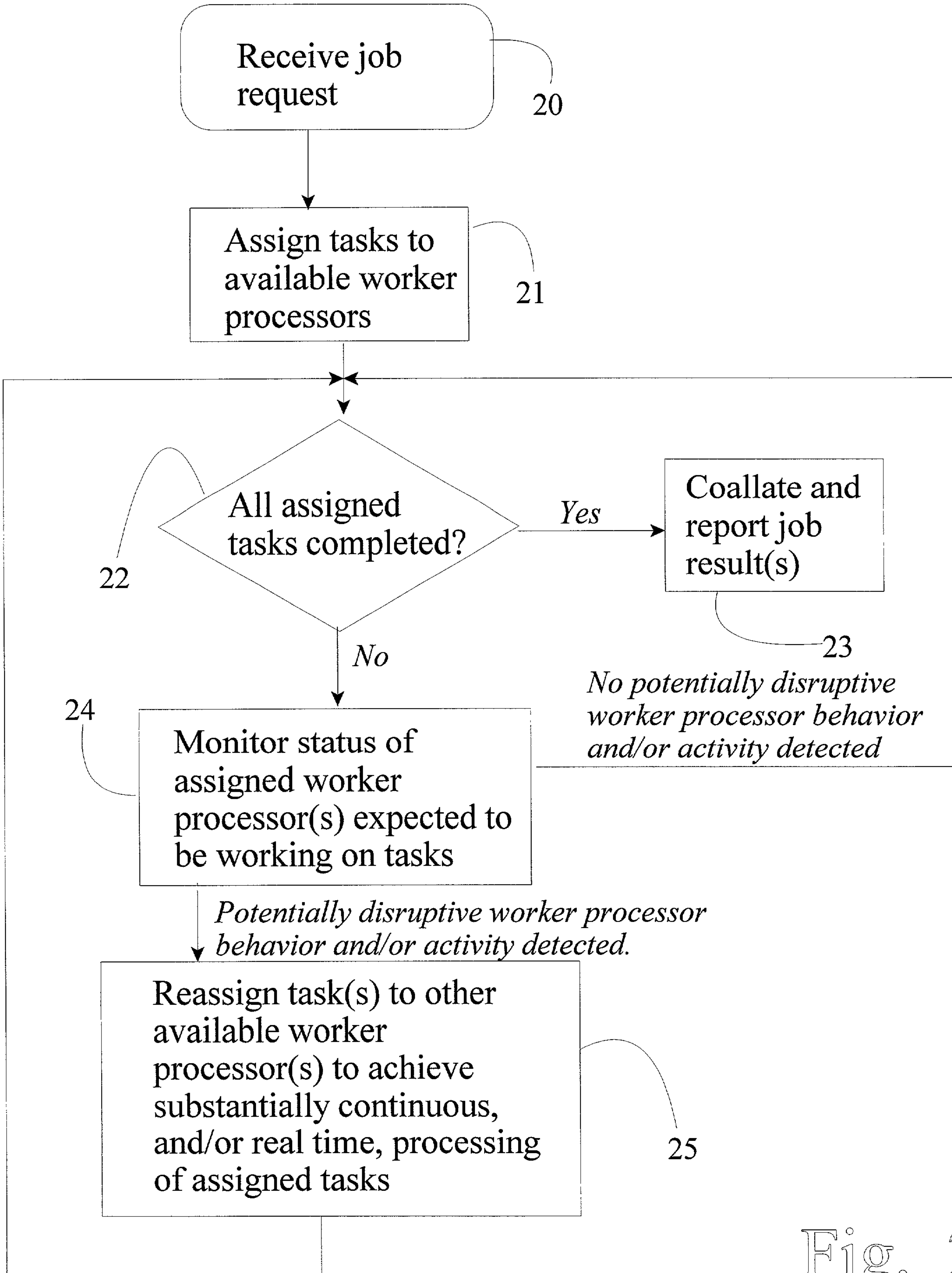
Receive job request
20
Assign tasks to available worker processors
21
All assigned tasks completed?
22
Yes
Coallate and report job result(s)
23
No
No potentially disruptive worker processor behavior and/or activity detected
24
Monitor status of assigned worker processor(s) expected to be working on tasks
Potentially disruptive worker processor behavior and/or activity detected.
Reassign task(s) to other available worker processor(s) to achieve substantially continuous, and/or real time, processing of assigned tasks
25

Fig. 2

Please type a plus sign (+) inside this box → ☐

PTO/SB/01 (10-00)
Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION (37 CFR 1.63)

☐ Declaration Submitted with Initial Filing OR ☐ Declaration Submitted after Initial Filing (surcharge (37 CFR 1.16 (e)) required)

Attorney Docket Number	1939-002
First Named Inventor	Bernardin
COMPLETE IF KNOWN	
Application Number	/
Filing Date	11/13/00
Group Art Unit	
Examiner Name	

As a below named inventor, I hereby declare that:

My residence, mailing address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Methods, Apparatus and Articles-of-Manufacture for Providing Always-Live Distributed Computing

(Title of the Invention)

the specification of which

☒ is attached hereto

OR

☐ was filed on (MM/DD/YYYY) ____ as United States Application Number or PCT International Application Number ____ and was amended on (MM/DD/YYYY) ____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the national or PCT international filing date of the continuation-in-part application.

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

☐ Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto:

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)	
09/583,244	05/31/00	☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

DECLARATION — Utility or Design Patent Application

Direct all correspondence to: ☐ Customer Number or Bar Code Label [ ] OR ☒ Correspondence address below

Name David Garrod, Ph.D., Esq.			
Address Hopgood, Calimafde, Judlowe & Mondolino, LLP			
Address 60 E. 42nd St.			
City NY	State NY	ZIP 10165	
Country USA	Telephone 212-551-5000	Fax 212-949-2795	

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

NAME OF SOLE OR FIRST INVENTOR : ☐ A petition has been filed for this unsigned inventor

Given Name (first and middle [if any]) James		Family Name or Surname Bernardin	
Inventor's Signature			Date
Residence: City Brooklyn	State NY	Country USA	Citizenship USA
Mailing Address			
Mailing Address			
City	State	ZIP	Country

NAME OF SECOND INVENTOR: ☐ A petition has been filed for this unsigned inventor

Given Name (first and middle [if any]) Peter		Family Name or Surname Lee	
Inventor's Signature			Date
Residence: City New York	State NY	Country USA	Citizenship USA
Mailing Address			
Mailing Address			
City	State	ZIP	Country

☐ Additional inventors are being named on the ____supplemental Additional Inventor(s) sheet(s) PTO/SB/02A attached hereto.